

CLAIMS

1. A method for determining the operational characteristics of a program, characterised in that it comprises a verification procedure comprising

5 the following steps:

- a first step comprising:

- expressing the operational characteristics of the program as functions dealing with occurrences or sequences of occurrences of events which may occur during possible executions of the program, said events being
10 able to deal with particular operations, particular values, at particular program points and in particular states of the program;
- determining a possible level of precision with which these characteristics must be determined;
- determining a possible set of particular contexts of execution in which
15 the program will always be executed;
- determining possible operational specificities of a set of platforms on which the program will be executed;

- a second step of estimation, by program analysis, and in consideration of said
20 possible level of precision, of said possible set of particular contexts of execution and of said possible operational specificities of platforms, of information relating to the structure of the program, the possible execution paths of the program and to the values of possible data, at various points of the

execution paths and under different execution conditions, of the states and data handled by the program;

- a third step for determining said operational characteristics, by means of the information extracted by said program analysis, by the computation of said functions on the occurrences or particular sequences of occurrences of particular operations, dealing with particular values, at particular points of the program, in particular states of the program, for the set of execution paths determined by analysis.

10

2. The method according to claim 1, characterised in that, in the case when the program is interactive and may depend on an undetermined number of dynamic values resulting from this interaction, the execution contexts are given by a description abstracted from the possible series of data representing said dynamic values.

15

3. The method according to claim 1, characterised in that, in the case where the program is inserted into a framework of execution, the static analysis also take into account the semantics of this execution framework, including the possible implicit interaction loops of the program.

20

4. The method according to claim 1, characterised in that certain of said particular operations (which form events, accompanied by constraints on the

values handled, the execution points, and the statuses of the program,) are defined as one of the following actions: call to a given routine, access to a given variable, reading or writing on a given port, computation of a given arithmetic expression, completion of execution of the program or of a routine
5 (on a normal return or ending an exception).

5. The method according to claim 1, characterised in that certain of said static analysis consist of abstract interpretations of the program, on abstract domains which may notably represent possible sets of values and symbolic
10 expressions.

6. The method as claimed in claim 1, characterised in that said extracted information are represented by means of one or more of the following structures: status graph of the program, inheritance graph, graph of the routine
15 calls of the program, control flow chart of each routine of the program, structure of loops and catch-up of exceptions, structure of basic blocks, abstraction of the status of the program at an execution point.

7. The method according to claim 1, characterised in that said
20 extraction of information does not apply to unnecessary information for determining the operational characteristics, both from the viewpoint of the amount of information extracted and from the precision of these pieces of information.

8. The method according to claim 1, characterised in that only the major pieces of information among said extracted information are computed and saved and in that the other pieces of information are only computed when
5 necessary for determining said operational characteristics.

9. The method according to claim 8, characterised in that the major pieces of information are information extracted at the breakdown nodes of the code of the routines in a graph of basic blocks and in that the other pieces of
10 information (in the body of the basic blocks) are recomputed by local analysis from information saved at the start and end of the corresponding block.

10. The method according to claim 1, characterised in that said operational characteristics represent validity criteria and in that said
15 determination establishes that the program is valid (because it observes each of said criteria), or invalid (because at least one of said criteria cannot be observed).

11. The method according to claim 10, characterised in that said
20 validity criteria express security or interoperability rules.

12. The method according to claim 1, characterised in that said operational characteristics characterise the resources which are consumed and

the functionalities which are exploited by the program during its execution and in that said determination provides an execution profile of the program.

13. The method according to claim 1, characterised in that the
5 computation of certain of said functions associated with the operational characteristics is performed during said static program analysis, as soon as certain of said pieces of information are extracted.

14. Application of the method according to claim 10 for automatic
10 filtering of a set of programs relative to a given set of validity criteria, characterised in that the extraction of information by static program analysis is only completed once per program and reused whenever necessary for determining whether the program observes said set of validity criteria.

15 15. A system for distribution of applications ensuring that the applications observe validity criteria associated with the execution platforms of these applications, characterised in that it comprises filtering means designed such that, for any client desiring to accede to the applications for a certain execution platform, the applications are filtered by a verification
20 procedure in accordance with the method according to any one of claims 1 to 12, only the applications which observe the validity criteria for said platform being presented to the client.

16. A system for multi-application execution ensuring that the applications observe given validity criteria, characterised in that it comprises:

- an application analysis server, a server for validation of applications and a multi-application platform, and

5 - means for ensuring, prior to loading or execution of an application on the platform:

- observance by this application of said criteria according to the method according to any one of claims 1 to 12, the extraction of information being carried out on the application analysis server and the evaluation of said criteria

10 being carried out on the server for validation of applications, and

- in the case when one of the criteria cannot be observed, the failure of the loading or execution of the application, the change of the status of the system and emission of a sound or visual signal to alert of failure of loading or execution.

15

17. The system according to claim 16, characterised in that the server for validation of applications is executed on the multi-application platform, the application analysis server executing outside the platform.

20

18. The system according to claim 16, characterised in that the application analysis server and the server for validation of applications are executed on the multi-application platform.